

```

// SPDX-License-Identifier: MIT
pragma solidity ^0.8.4;
import "@openzeppelin/contracts/token/ERC721/ERC721.sol";
import
"@openzeppelin/contracts/token/ERC721/extensions/ERC721Enumerable.sol";
import
"@openzeppelin/contracts/token/ERC721/extensions/ERC721Burnable.sol";
import "@openzeppelin/contracts/access/Ownable.sol";
/// @custom:security-contact
whitefeather@whitebuffalotribe.onmicrosoft.com
contract Draconian is ERC721, ERC721Enumerable, ERC721Burnable, Ownable {
    constructor() ERC721("Draconian", "DGN") {}
    function safeMint(address to, uint256 tokenId) public onlyOwner {
        _safeMint(to, tokenId);
    }
    // The following functions are overrides required by Solidity.
    function _beforeTokenTransfer(address from, address to, uint256
tokenId)
        internal
        override(ERC721, ERC721Enumerable)
    {
        super._beforeTokenTransfer(from, to, tokenId);
    }

import express from 'express';
import { collectDefaultMetrics, register } from 'prom-client';
collectDefaultMetrics();
const app = express();
app.get('/metrics', async (_req, res) => {
    try {
        res.set('Content-Type', register.contentType);
        res.end(await register.metrics());
    } catch (err) {
        res.status(500).end(err);
    }
});
app.listen(4001, '0.0.0.0')
    function supportsInterface(bytes4 interfaceId)
        public
        view
        override(ERC721,
ERC721(Enumerable)
        returns (bool)
    {
        return super.supportsInterface(interfaceId);
    }
}

```